Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 7, No. 3, 2023

PCB Defect Detection Algorithm Based on Improved YOLOv8

Yan Long[1, *], Zhiang Li[1], Yinan Cai[1], Rongwei Zhang[1], Ke Shen[1]

[1] School of Computer Science, Yangtze University, Jingzhou, 434023, China

[*] Corresponding author: Yan Long (Email: 2022710657@yangtzeu.edu.cn)

Abstract: In order to solve the performance and efficiency problems in PCB defect detection tasks, a PCB defect detection algorithm based on improved YOLOv8 is proposed, which aims to improve detection accuracy, reduce model complexity, adapt to small target detection, and operate in resource-constrained environments. achieve efficient detection. First, an improved neck network structure is introduced, which reduces the number of parameters and computational complexity of the model and improves resource utilization. Subsequently, the ShuffleAttention and BiFPN structure were added to enhance the model's multi-scale feature fusion capabilities and better adapt to small target detection. Finally, the WIoU loss function is used to replace the traditional CIoU loss function, thereby improving the detection accuracy and robustness of the model. Experimental results show that the improved algorithm achieved 94.2% and 49.0% in mAP50 and mAP90-95 respectively. The number of parameters, GFLOPs and weight size of the model were reduced by 33%, 12% and 32% respectively, reaching 1.992M, 7.1 and 4.2M. Provides an efficient, accurate and lightweight PCB defect detection solution, making it ideal for resource-constrained environments such as mobile devices and embedded systems.

Keywords: BiFPN, PCB defect detection, ShuffleAttention, WIoU, YOLOv8.

1. Introduction

In modern electronics manufacturing, printed circuit boards (PCBs) play a vital role as one of the core components of electronic equipment. However, due to its highly complex manufacturing process and the impact of the external environment and operating errors, PCBs often suffer from defects, such as short circuits, missing holes, burr problems, etc. These defects can cause performance degradation or even complete failure of electronic devices, so efficient and accurate defect detection of PCBs is crucial. In recent years, deep learning technology has made great breakthroughs in image processing and computer vision, providing a new solution for PCB defect detection.

There are currently two main target detection algorithms. One is a two-stage detection algorithm, such as Faster RCNN, Cascade RCNN, and Mask RCNN. Hu et al. [1] built a new network based on Faster RCNN to detect PCB defect images, which can more accurately detect small defects on PCB. Ding et al. [2] proposed a tiny defect detection network TDD-Net based on Faster RCNN. The results showed that the mAP in PCB defect detection reached 98.0%. The other is a single-stage detection algorithm, such as YOLO series, SSD, etc. Wang et al. [3] proposed a PCB plug-in solder joint defect detection method based on YOLOv3 that combines ordered probability density weighting and attention mechanisms. After testing, the average accuracy of the improved method can reach 96.69% and is better than the original network. Better convergence. Liao et al. [4] proposed an algorithm based on improved YOLOv4 for PCB surface defects. Experimental results show that the average accuracy reaches 98.6%, reducing the amount of model parameters and calculations and achieving high performance. Tang et al. [5] proposed a PCB surface defect detection algorithm PCB-YOLO based on YOLOv5. By introducing the Swin Transformer and K-means++ algorithm, PCB-YOLO achieved a good balance between performance and consumption. mAP reached 95.97% at 92.5 FPS. Chen et al. [6] proposed a PCB defect detection algorithm based on improved YOLOv7. Using FasterNet as the backbone network combined with the CBAM attention mechanism can effectively solve the problems of slow speed and low accuracy in the PCB defect detection process.

However, the above method models may be too complex, require high computing resources, and are unsuitable for deployment on lightweight devices and mobile devices, limiting their potential in actual PCB defect detection applications. To overcome this limitation, this study aims to improve the YOLOv8 model to make it more suitable for PCB defect detection applications. Specifically, the goal is to achieve lightweight models to reduce model volume and computational requirements while improving detection performance to ensure efficient deployment in limited resource environments such as mobile devices. The main contributions of this article are as follows:

(1) By improving the neck network of the original YOLOv8 while reducing the number of parameters, the detection accuracy of the model is also enhanced.

(2) The BiFPN [7] is introduced into the improved neck network so that the neck network can more effectively integrate multi-scale feature networks and improve the ability to identify small target defect features.

(3) Add the ShuffleAttention (SA) [8] to the improved neck network to improve the model's ability to locate small targets and reduce the false detection rate and missed detection rate.

(4) Replace CIoU with WIoU loss function [9] to improve the quality of anchor boxes, reduce the difficulty of detecting small targets, and improve the detection accuracy of defect features of small targets.

2. Improve YOLOv8 Network Structure

2.1. Improved YOLOv8 network

YOLOv8 is one of the latest models of the YOLO series [10-14]. Although the network structure of YOLOv8 is similar to that of YOLOv5, YOLOv8 has made remarkable achievements in object detection through some unique improvements and performance optimization. Firstly, the backbone network still adopts the idea of CSP but replaces the C3 module in YOLOv5 with the C2f module to achieve further lightweight and adjusts the number of channels for different scale models, significantly improving the model's performance. At the same time, the SPPF module used in architectures such as YOLOv5 is still used. The neck network still uses the idea of PAN, but the convolution structure of CBS1×1 in the up-sampling stage of PAN-FPN in YOLOv5 is deleted. The head network changed from the original coupling head to the decoupling head, separating the classification head and the detection head, and at the same time changed from Anchor-Based to Anchor-Free. The TaskAlignedAssigner matching strategy is used for Loss calculation, and the DFLoss loss function is introduced. YOLOv8 can be used in various application fields, including intelligent security, automatic driving, object tracking, industrial quality inspection, etc. Its efficient performance makes it ideal for many computer vision projects.

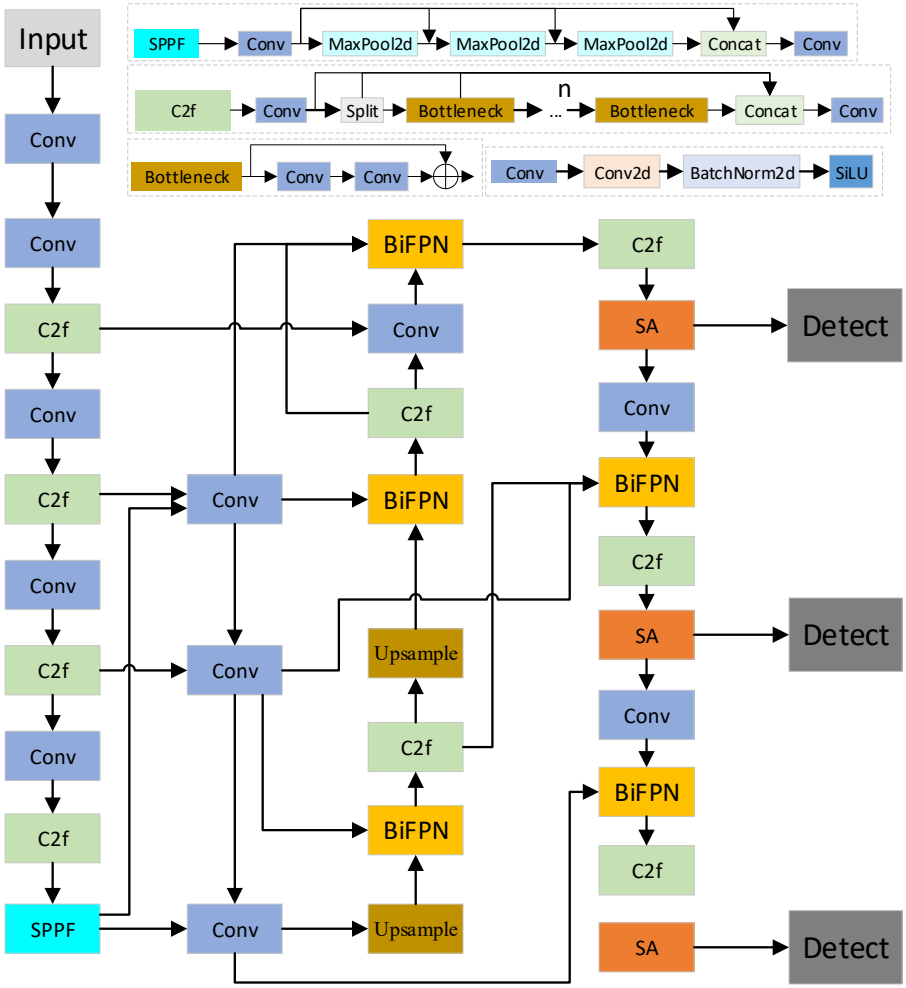

Figure 1. Improved YOLOv8 structure diagram

This article further improves the YOLOv8 network model, as shown in Figure 1. It first introduces an improved neck network structure to reduce the number of parameters and calculations of the model. Then, the SA attention mechanism and BiFPN structure are added to the neck network to improve the multi-scale feature fusion ability of the model. Finally, CIoU is replaced by the WIoU loss function to speed up the convergence of the model and improve the overall detection accuracy of the model.

2.2. ShuffleAttention

There are two main types of attention mechanisms, one is spatial attention, and the other is channel attention. CBAM integrates these two attention mechanisms and can achieve better results by capturing the dependencies between channels and the pixel-level relationships in space. However, while improving accuracy, it also brings more calculations. The SA attention mechanism divides the spatial and channel attention mechanisms into blocks and uses them in parallel to combine them efficiently, improving accuracy without causing more additional calculations.

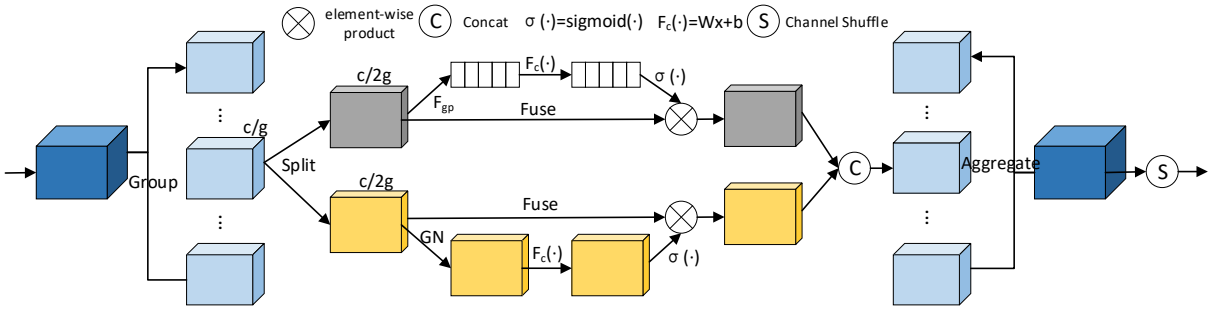

Figure 2. SA attention mechanism structure diagram

As shown in Figure 2, the SA attention mechanism first divides the features into g groups, and a part of each group uses the channel attention mechanism, as shown in the gray part of the figure. The specific implementation is similar to the SE attention mechanism[15]; another Part of it uses the spatial attention mechanism, as shown in the yellow part of the figure, where GN represents the group normalization operation. Then the two parts are superimposed according to the number of channels to achieve information fusion. Finally, a random mixing operation is performed on g groups to obtain the final output feature map. In current PCB defect detection tasks, due to the small size of defect features, traditional inspection methods do not perform well in handling these small-sized features because they are difficult to capture and emphasize. In order to solve this problem, this article decided to introduce the SA attention mechanism to improve the detection accuracy of PCB defects without affecting the amount of calculation.

2.3. BiFPN structure

The PANet module [16] is utilized in the neck network of the original YOLOv8 model. As depicted in Figure 3, PANet combines features of various scales through top-down information transfer, which requires significant computational resources. In terms of complexity, PANet can be more resource-intensive compared to lightweight methods, resulting in slower training and inference speeds. Moreover, due to potential loss of detailed information for small targets from the top layer during top-down information transfer, feature extraction capabilities for small targets may be compromised. Additionally, PANet typically necessitates storing and processing numerous feature maps, particularly when handling high-resolution images; this could limit its application in resource-constrained environments.

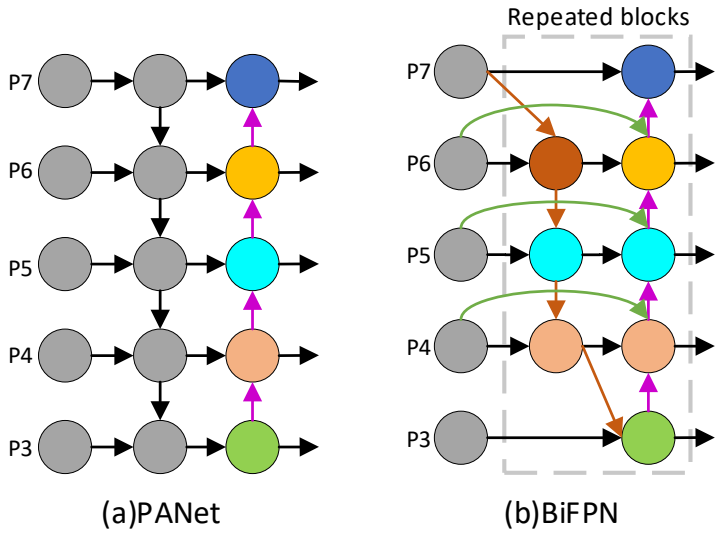

Figure 3. Diagrams of PANet and BiFPN structures

This article replaces the PANet module with the BiFPN module. The BiFPN structure is shown in b in Figure 3. First, BiFPN introduces a two-way information flow, transferring features freely between different levels. This bidirectionality helps to capture multi-scale features better, especially when dealing with small or large targets. Then, the structure of BiFPN is relatively lightweight and usually requires fewer computing resources. This is useful for object detection tasks embedded in lightweight models or mobile devices, providing better speed and efficiency. Finally, due to the bi-directionality of BiFPN, gradients can be propagated back to the underlying feature map more smoothly, helping to train more profound and complex models. This helps speed up model convergence and improves training stability.

2.4. Improved neck network

The design of the original neck network is usually optimized for a specific task and data set, and the performance of the neck network may be limited if applied to a domain or data set that is different from the original task. The YOLOv8 original neck network usually introduces an additional computational burden because it requires further processing of the feature maps extracted by the backbone network. This can cause models to run slower when computing resources are limited, especially in embedded devices or mobile applications. At the same time, the design of the neck network

may not be suitable for very small or substantial target objects because the information on the feature map may lose details or fail to capture the global characteristics of the target. This paper improves and optimizes the neck network for PCB small target defect detection.

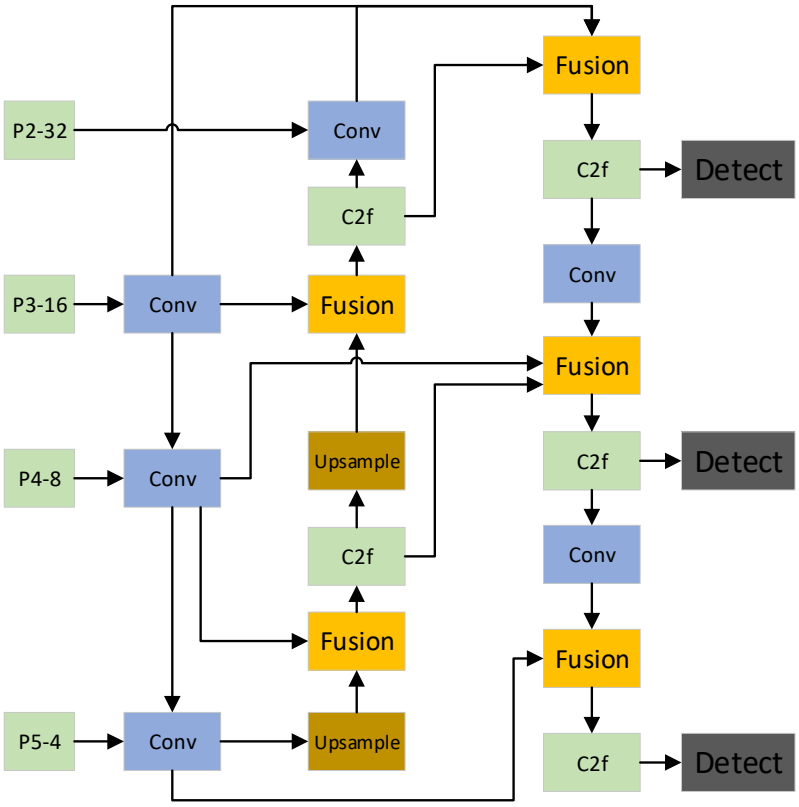

Figure 4. Improved neck network structure diagram

As shown in Figure 4, first, three convolutional layers are added after the backbone network, and the number of channels in the neck network is fixed to 256 so that the network focuses more on the quality of feature fusion rather than the quantity while reducing the network Memory footprint and computational overhead. Then, the p2 part in the backbone network is added to the feature fusion of the neck network because it contains more detailed information, which can help the network detect smaller targets. The features of P2 can be used to locate the bounding box of the target more accurately, which helps to improve the localization accuracy of the target. Finally, three more layers are added to the neck network, namely Conv, Fusion, and C2f, to improve the performance and detection accuracy of the model. Among them, the parameters in Fusion are Weight, Adaptive, Concat, and BiFPN. This article uses the BiFPN fusion method[17].

2.5. WIou Loss

The original YOLOv8 model uses the CIoU bounding box loss function. The CIOU (Complete Intersection over Union) loss function calculates the bounding box (Bounding Box) prediction quality in the target detection task. It is an extension of IoU (Intersection over Union) and aims to measure the degree of overlap between two bounding boxes accurately.

The traditional IoU loss function calculates the intersection and union of two bounding boxes and then uses their ratio to measure the overlap between the predicted and true bounding boxes. However, there are some problems with IoU, especially when the height or width of the bounding box is close to zero, which will lead to inaccurate loss gradients, which is not conducive to model training. The CIOU loss function improves IoU by considering the relationship between center point distance, width, and height between bounding boxes. Its calculation formula is as follows:

$$L_{CIOU} = 1 - IOU + \frac{p^2(b, b^{gt})}{c^2} + \alpha\upsilon \tag{1}$$

In the formula: IOU is the traditional intersection and union ratio, p^2 is the square of the Euclidean distance between the predicted bounding box and the center point of the true bounding box, c^2 is a factor used to normalize the center point distance, equal to the diagonal of the true bounding box Squared, α is a weight parameter used to balance the influence of center point distance, and υ is a term measuring the difference between the width and height of the true and predicted bounding boxes.

Although CIOU shows improvement in small target detection, there is still a certain instability for targets with extremely small or extreme aspect ratios. WIOU uses a dynamic non-monotonic focusing mechanism to evaluate the quality of the anchor box and provides an allocation strategy for the gradient gain of the bounding box to enhance the target positioning capability of the algorithm. WIOU constructs a two-layer attention mechanism to enable the model to perform better generalization. WIOUv1 is defined as follows:

$$L_{WIOUv1} = R_{WIOU} L_{IOU} \tag{2}$$

In the formula:

$$R_{WIOU} = \exp(\frac{(x-x_{gt})^2+(y-y_{gt})^2}{(W_g^2+H_g^2)^*}) \quad (3)$$

$$L_{IOU} = 1 - IOU \quad (4)$$

Among them, $W_g^2$and H_g^2 represent the width and height of the minimum anchor box, which are used to generate gradients that hinder convergence. WIOUv3 introduces the dynamic non-monotone focusing coefficient r on the basis of WIOUv1. Its formula is defined as follows:

$$L_{WIOUv3} = rL_{WIOUv1} \quad (5)$$

$$r = \frac{\beta}{\delta\alpha^{\beta-\delta}} \quad (6)$$

$$\beta = \frac{L_{IOU}^*}{\overline{L_{IOU}}} \in [0,+\infty) \quad (7)$$

In the formula: L_{IOU}^* represents the separation of L_{IOU} from the calculation graph, $\overline{L_{IOU}}$ is the running average of the momentum m, β is the outlier, α and δ are the hyperparameters that control the gradient gain r.

3. Experimental Results and Analysis

3.1. Dataset Introduction

The PCB defect data set used in this paper is from the Open Laboratory of Intelligent Robotics of Peking University. There are 693 images and 6 defect types, including missing hole, mouse bite, open circuit, short, spur, and spurious copper. The dataset was divided into a training set and a validation set according to the ratio of 8:2, and the training was carried out by online data augmentation. Figure 5 shows the magnified images of various defects, and each image contains 3 to 5 such target defects.

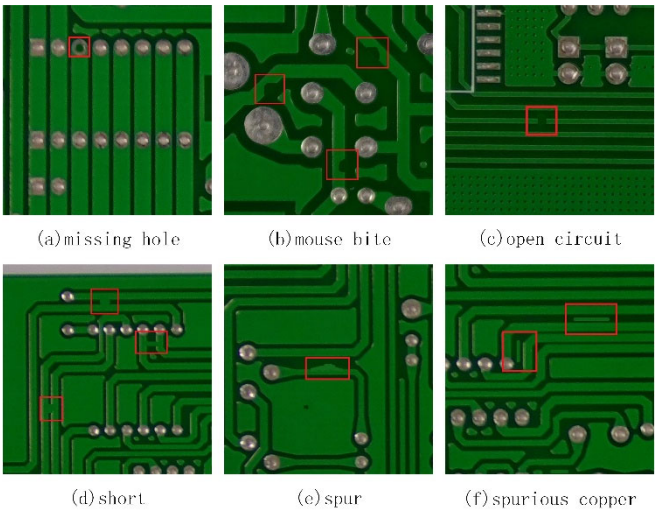

Figure 5. PCB defect example diagram

3.2. Experimental environment configuration

The experimental hardware environment configuration of this article is NVIDIA GeForce RTX 3080, 12G video memory, and the compilation environment is Python3.8.16+torch2.0.0+CUDA11.8. Training is performed in the PyTorch network framework. The optimizer uses SGD. The hyperparameters are set to a batch size of 32, a training period of 200, an initial learning rate of 0.01, and no pre-training weights are added during the training process.

3.3. Evaluation index

The evaluation indicators used in this article include precision P, recall R, F1 factor, average precision mAP, parameter amount Params, Giga floating point budget GFLOPs per second, and model weight Weights. The relevant formulas are as follows:

$$P = \frac{TP}{(TP+FP)} \quad (8)$$

$$R = \frac{TP}{TP+FN} \quad (9)$$

$$F1 = \frac{2\times(P\times R)}{(P+R)} \quad (10)$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n}\int_0^1 \mathrm{P(R)}d(\mathrm{R}) \quad (11)$$

In the formula, TP represents the number of true examples, FP represents the number of false positive examples, and FN represents the number of false negative examples.

Among them, the precision rate measures the accuracy of the model in positive category prediction; the recall rate measures the model's coverage in positive category samples; the F1 factor is used to measure the performance of the model comprehensively; the average accuracy rate calculates the accuracy of the model in different categories. The average accuracy value is then averaged; the number of parameters refers to the number of trainable parameters in the model. Generally, the more parameters, the more complex the model; GFLOPs represent the billions of floating point operations performed by the model per second, and The speed of the model is related to performance; the size of the model weight is associated with the storage requirements and deployment costs of the model. The smaller the weight, the more suitable it is for facilities such as mobile devices.

3.4. Result analysis

3.4.1. Loss function comparison experiment

In order to verify the effectiveness of replacing the CIoU loss function with the WIoU loss function in the original YOLOv8 model, this article compares some mainstream loss functions, such as DIoU [18], EIoU [19], GIoU [20], SIoU [21] and FocalIoU [22], the results are shown in Table 1.

Table 1. Loss function comparison experiment

Loss function	mAP50(%)	mAP90-95(%)	P(%)	R(%)	F1(%)
CIoU	90.8	46.8	91.8	85.4	88.0
DIoU	91.5	47.3	93.9	86.5	90.0
EIoU	90.7	47.5	92.8	84.8	89.0
GIoU	89.9	45.8	90.6	85.4	88.0
SIoU	90.2	47.1	90.6	85.7	88.0
FocalIoU	91.7	46.9	92.9	86.7	90.0
WIoU	93.0	49.1	91.4	89.8	91.0

The results show that the WIoU loss function shows excellent advantages on multiple key performance indicators. Specifically, the highest performance is achieved regarding mAP, and defective targets can be found more accurately. It

also leads to other loss functions regarding precision and recall, showing that it can better predict true examples and detect more actual positive ones. The F1 factor description of WIoU shows significant superiority in this paper's PCB defect detection task.

According to the line chart in Figure 6, we can clearly observe the impact of different IOU calculation methods on the performance of the YOLOv8 model. In terms of val/box_loss, WIoU shows obvious advantages, and its loss value is relatively low, indicating that WIoU has a better effect in object boundary box matching.

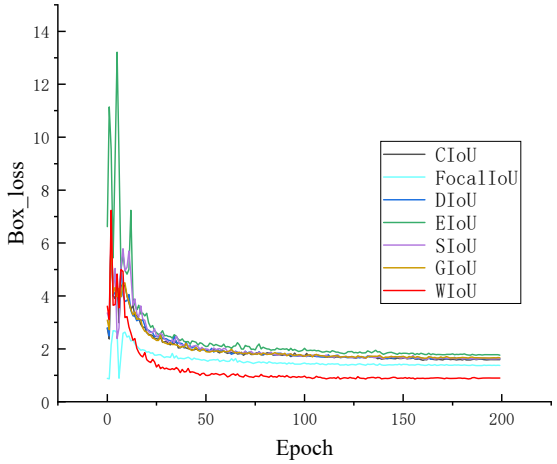

Figure 5. Experimental results of different loss functions

3.4.2. Ablation experiment

In order to verify the effectiveness of this article's improvement of the YOLOv8 network structure, ablation experiments were conducted on different improvement points under the same conditions in the experimental environment. The results are shown in Table 2. Improvement 1 is the introduction of the neck network improved in this article. The results show that the accuracy is significantly improved compared to the original YOLOv8n. At the same time, the number of parameters and GFLOPs of the model have decreased, and the model weight has also become smaller. Improvement 2, Improvement 3 and Improvement 4 add BiFPN, SA attention mechanism and WIoU to the original YOLOv8 network respectively. It can be seen that mAP has improved when the number of parameters and GFLOPs are not much different. Improvement 5 is that after fusing the improved neck network with BiFPN, the number of parameters, GFLOPs and weight file size of the model are lower than those of the original YOLOv8n, Improvement 1 and Improvement 2. Although mAP is lower than Improvement 1, after adding the SA attention mechanism through Improvement 6 without affecting the number of parameters, GFLOPs and weight file size of the model, mAP is still higher than the previous experiment. Finally, the algorithm in this article integrates four improvement points into the original YOLOv8n. It can be seen that mAP50 has increased by 3.4%, and the number of parameters, GFLOPs, and weight file sizes have been reduced by 33%, 12%, and 32% respectively.

Table 2. Loss function comparison experiment

Model	New Neck	BiFPN	SA	WIoU	mAP50(%)	Params/M	GFLOPs	Weight/M
YOLOv8n					90.8	3.007	8.1	6.2
Improvement 1	√				93.2	2.071	7.3	4.4
Improvement 2		√			91.0	3.068	8.3	6.4
Improvement 3			√		91.6	3.007	8.1	6.3
Improvement 4				√	93.0	3.007	8.1	6.2
Improvement 5	√	√			92.8	1.992	7.1	4.2
Improvement 6	√	√	√		93.7	1.992	7.1	4.2
Ours	√	√	√	√	94.2	1.992	7.1	4.2

3.4.3. Comparative Experiment

In order to verify the effectiveness of this algorithm, a comparative experiment as shown in Table 3 was established. The experimental results show that it can be clearly seen that the algorithm in this paper has shown advantages in many aspects. First of all, from the perspective of the number of model parameters, it is more lightweight than other models. The number of parameters is only 1.992M, which is far lower than the 24.280M of SSD[23], which makes it more suitable for deployment in mobile devices and other resource-limited environments. environment. In addition, the computational load of the algorithm in this paper is also relatively low, with GFLOPs of only 7.1, which is far lower than the highest value of 275.4 in other models, which means that it can achieve efficient real-time detection on mobile devices.

Not only that, the algorithm in this paper has also significantly improved the detection accuracy. Its mAP50 value is 94.2%, which is higher than most other models, even close to YOLOv8s' 94.7%, and it also performs well in mAP90-95, reaching 49.0%. This means that the algorithm in

this paper can more accurately capture the target object when detecting the target and provide more reliable detection results.

Table 3. Loss function comparison experiment

Model	mAP50(%)	mAP90-95(%)	Params/M	GFLOPs	Weight/M
SSD	84.5	36.5	24.280	275.4	93.1
YOLOv5n	65.9	27.7	1.767	4.2	3.9
YOLOv5s	87.4	41.6	7.026	15.8	14.5
YOLOv6s	90.3	45.5	16.298	44.0	32.8
YOLOv7-tiny	79.4	35.5	6.028	13.2	12.3
YOLOv8n	90.8	46.8	3.007	8.1	6.2
YOLOv8s	94.7	51.2	11.128	28.4	22.5
Ours	94.2	49.0	1.992	7.1	4.2

In summary, the algorithm in this paper has become an ideal choice for deployment in resource-limited environments such as mobile devices due to its lightweight characteristics, low computational load, and superior detection accuracy. This gives it broad application potential in areas such as real-time object detection, mobile applications, and embedded systems.

3.4.4. Visual result analysis

In order to verify the superiority of this algorithm in actual detection, five sets of experimental comparisons are designed as shown in Figure 7, all of which are enlarged images. First of all, the first set of results shows that our algorithm surpasses the original YOLOv8n in terms of detection accuracy, which means that our algorithm identifies defects more accurately. Secondly, the second and third sets of results show that compared with the original YOLOv8n, the missed detection rate of this algorithm has been reduced, which means that it is easier to capture potential defects, thus improving the comprehensiveness of detection. Finally, the fourth and fifth sets of results show that the algorithm in this paper has reduced the false detection rate, which means that normal objects are less likely to be misjudged as defects, thereby reducing the risk of false alarms. Taken together, the algorithm in this paper has achieved significant improvements in accuracy, missed detection rate, and false detection rate, making it a more reliable and efficient defect detection tool.

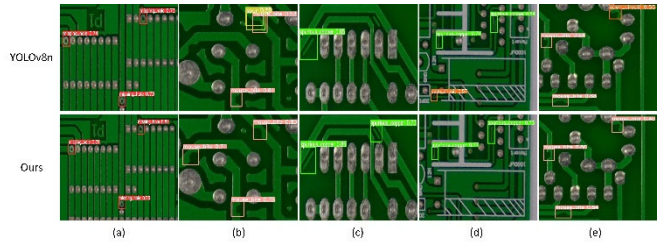

Figure 7. Visualized results comparison chart

4. Conclusion

This study proposes an improved YOLOv8 model for PCB defect detection through a series of improvements and optimizations. In the experiment, this paper first introduced a new neck network structure, which effectively reduced the number of parameters and computational complexity of the model, thereby achieving higher operating efficiency in resource-constrained environments. Subsequently, the SA (ShuffleAttention) attention mechanism and BiFPN structure were added to further improve the model's multi-scale feature fusion capabilities, making it more suitable for small target detection tasks. Most importantly, replacing the traditional CIoU loss function with the WIoU loss function significantly improves the detection accuracy and robustness of the model.

Experimental results verify the excellent performance of this improvement in PCB defect detection tasks. The model in this article not only achieves significant improvements in mAP50 and mAP90-95, reaching 94.2% and 49.0% respectively, but also has a relatively low number of parameters and calculations, making it suitable for mobile devices and embedded systems. Ideal for resource-constrained environments such as systems. By comparing other common target detection models, this method achieves a good balance between lightweight and performance, and has broad application prospects.

Taken together, this research provides an efficient, accurate and lightweight solution for the field of PCB defect detection, which has important practical significance for applications such as industrial quality inspection and automated production. Future work can further explore the applicability of the model in wider areas and investigate how to further improve detection performance and efficiency.

References

[1] Hu B, Wang J. Detection of PCB surface defects with improved faster-RCNN and feature pyramid network[J]. IEEE Access, 2020, 8: 108335-108345.

[2] Ding R, Dai L, Li G, et al. TDD-net: a tiny defect detection network for printed circuit boards[J]. CAAI Transactions on Intelligence Technology, 2019, 4(2): 110-116.

[3] Wang Z, Chen W, Li T, et al. Improved YOLOv3 detection method for PCB plug-in solder joint defects based on ordered probability density weighting and attention mechanism[J]. AI Communications, 2022 (Preprint): 1-16.

[4] Liao X, Lv S, Li D, et al. YOLOv4-MN3 for PCB surface defect detection[J]. Applied Sciences, 2021, 11(24): 11701.

[5] Tang J, Liu S, Zhao D, et al. PCB-YOLO: An improved detection algorithm of PCB surface defects based on YOLOv5[J]. Sustainability, 2023, 15(7): 5963.

[6] Chen B, Dang Z. Fast PCB defect detection method based on FasterNet backbone network and CBAM attention mechanism integrated with feature fusion module in improved YOLOv7[J]. IEEE Access, 2023.

[7] Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.

[8] Zhang Q L, Yang Y B. Sa-net: Shuffle attention for deep convolutional neural networks[C]//ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021: 2235-2239.

[9] Tong Z, Chen Y, Xu Z, et al. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism[J]. arXiv preprint arXiv:2301.10051, 2023.

[10] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[11] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[12] Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021.

[13] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv:2209.02976, 2022.

[14] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 7464-7475.

[15] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[16] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.

[17] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.

[18] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.

[19] Zhang Y F, Ren W, Zhang Z, et al. Focal and efficient IOU loss for accurate bounding box regression[J]. Neurocomputing, 2022, 506: 146-157.

[20] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 658-666.

[21] Gevorgyan Z. SIoU loss: More powerful learning for bounding box regression[J]. arXiv preprint arXiv:2205.12740, 2022.

[22] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[23] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.